



## **Aula 00 - Infraestrutura em TI**

*Tecnologia da Informação – TCE/RJ*

*(Analista de Controle Externo)*

**Prof. Rômulo Santos**

---

## Sumário

---

<b>SUMÁRIO.</b>	<b>2</b>
<b>APRESENTAÇÃO.</b>	<b>3</b>
<b>SISTEMAS OPERACIONAIS LINUX. CONCEITOS BÁSICOS. NOÇÕES DE ADMINISTRAÇÃO:</b>	<b>5</b>
HISTÓRIA DO GNU/LINUX.	5
DISTRIBUIÇÃO E CARACTERÍSTICAS.	18
SHELL.	19
FHS.	22
DISCOS E PARTIÇÕES.	25
MBR E GPT.	27
SISTEMAS DE ARQUIVOS.	30
<b>QUESTÕES COMENTADAS PELO PROFESSOR.</b>	<b>36</b>
<b>REFERÊNCIAS.</b>	<b>45</b>

## Apresentação

Olá concursando, é com imensa satisfação que anunciamos essa grande parceria entre o Direção Concursos e o Dominando TI. Caso ainda não me conheça, sou o professor Rômulo Santos e leciono disciplinas ligadas à área de Tecnologia da Informação, como ênfase em Microinformática, Sistemas de Computação, Arquitetura e Organização de Computadores, Sistemas Operacionais, Linux, Infraestrutura de TI, *Datacenters*, Bancos de Dados, Redes de Computadores, SGBD Oracle, dentre outras. Também me dedico a um Programa de Estudos de “Alto Rendimento para Concursos” que por meio de sucessivas evoluções, vem alcançando aprovações e resultados extremamente satisfatórios. Meu objetivo principal será o direcionamento da sua preparação em concursos públicos visando a obtenção de resultados ainda mais eficientes.

Minha trajetória acadêmica iniciou com a bacharelado em Análise de Sistemas (UFMS). Logo, em seguida concluí algumas especializações aos quais destaco Criptografia e Segurança em Redes (UFF), Aplicações Complementares às Ciências Militares (EsFCEEx), Gestão de Tecnologia da Informação na Administração Pública (JK) e Ciências Militares (EsAO). Tive a grata felicidade de defender o Mestrado em Ciência da Informação (Unb), e atualmente sou doutorando em Gestão de Projetos de TIC (UNINI).

Já na área profissional pude exercer várias funções ligadas à área tecnológica. Atualmente sou gerente de TI no Exército Brasileiro, instituição em que ingressei no ano de 1997. Tenho experiência na área de Ciência da Informação, com ênfase em Arquitetura da Informação, Organização e Recuperação da Informação e na área de Ciência da Computação, com ênfase em Metodologias e Técnicas da Computação, Gestão de Projetos de TI, Gestão de Equipes de Desenvolvimento de *Software*, Infraestrutura de *Datacenters*, Criptografia, Segurança da Informação, Bancos de Dados, *Business Intelligence* e *Data Warehouse*. Participei de projetos envolvendo desenvolvimento de Sistemas Corporativos, Banco de Dados, Certificação Digital, Telefonia VoIP, Segurança da Informação, BI, OLAP, ETL e DW.

Além disso, consegui algumas certificações de mercado, entre as quais, a *Microstrategy Certified Project Designer 9.2.1 (Microstrategy University)*, *Oracle Certified Associate (OCA) - Oracle Database Administrator (Oracle University)*, *Oracle Certified Professional (OCP) - Oracle Database Administrator (Oracle University)*, *Oracle Certified Expert (OCE) - Oracle Real Application Clusters Administrator (Oracle University)*.

Minha experiência em docência iniciou-se em 1998, no Exército Brasileiro, onde pude contribuir por 12 anos na capacitação, preparação e formação técnico-profissional de milhares de jovens. Nesse período pude ser condecorado com as comendas da “Ordem do Mérito Thomaz Coelho” e da “Ordem do Mérito Marechal Trompowsky”, ambas do Instituto dos Docentes do Magistério Militar. Desde 2010, leciono em cursos preparatórios para concursos públicos, nas modalidades *online* e presencial. Encontrei nessa atividade uma satisfação profissional jamais desfrutada.

Agora que nos conhecemos podemos passar para o mais interessante, ou seja, falar sobre esse nosso **curso**.

**Você está iniciando os estudos em TI para concursos?** Se sim, isso não será um problema. O curso o atenderá perfeitamente, uma vez que, será abordada “em detalhes” toda a teoria necessária para sua preparação, assim como, a resolução comentada de inúmeros exercícios para que você possa entender e praticar bastante cada aspecto estudado. E caso você já seja um pouco mais experiente terá uma ótima oportunidade para revisar minuciosamente os assuntos.

Inicialmente faça uma “leitura na diagonal” (*skimming*) desse material. O objetivo dessa fase é captar a “ideia geral” e identificar os assuntos a serem tratados. Em seguida, assista às videoaulas anotando as dúvidas que surgirem ao longo do processo. A terceira fase consiste em “estudar” novamente esse material com maior nível de detalhes e aprofundamento. Nessa etapa, não esqueça de realizar os resumos para futuras revisões. Por fim, faça uma maior quantidade de exercícios possíveis. Durante esse processo, fique à vontade para me procurar no fórum de dúvidas sempre que for necessário.

O curso contempla os seguintes materiais:



**Curso completo em videoaulas**  
*teoria e exercícios resolvidos sobre os pontos do edital*

**Curso completo escrito (pdf)**  
*teoria e mais exercícios resolvidos sobre os pontos do edital*

**Fórum de dúvidas**  
*para você sanar suas dúvidas sempre que necessário*

### Como este curso está organizado

Aula	Conteúdo do edital
00	Sistemas operacionais Linux. Conceitos básicos. Noções de administração (parte 1).
01	Sistemas operacionais Linux. Conceitos básicos. Noções de administração (parte 2).
02	Sistemas operacionais Linux. Conceitos básicos. Noções de administração (parte 3).
03	Tecnologias de backup. Deduplicação.
04	Tecnologias de armazenamento DAS, NAS e SAN.
05	Serviços de armazenamento, padrões de disco e de interfaces. RAID.
06	Virtualização.
07	Cloud Computing.
08	Topologia típica de ambientes com alta disponibilidade e escalabilidade. Balanceamento de carga, fail-over e replicação de estado.
09	Tecnologias e arquitetura de datacenter. ILM – Information Lifecycle Management. Interoperabilidade.

### Contatos

 [romulo@dominandoti.com.br](mailto:romulo@dominandoti.com.br)

 [@profromulosantos](https://www.instagram.com/profromulosantos)

 [/professorromulosantos](https://www.facebook.com/professorromulosantos)

# SO Linux. Conceitos básicos. Noções de administração.

Caro concursando,

Abordaremos inicialmente a História do GNU/LINUX. Esse tópico não foi abordado na videoaula e o considero como de "leitura complementar".

Posteriormente detalharemos vários assuntos, dentre os quais:

- O que é uma distribuição Linux e as características principais da distribuição Red Hat Linux;
- As características e funções do *Shell*;
- As funções dos diretórios que compõem a *Filesystem Hierarchy Standard* (FHS);
- Como o Linux trata os discos e partições;
- Diferenças entre o *Master Boot Record* (MBR) e o *GUID Partition Table* (GPT);
- Os gerenciadores de *boot*;
- As características dos sistemas de arquivos EXT<sub>2</sub>, EXT<sub>3</sub>, EXT<sub>4</sub> e ReiserFS;
- Como formatar uma partição de disco;
- Como montar e acessar uma partição de disco;
- Os comandos FSTAB, DF e DU.

Tenho certeza que nos divertiremos muito nas próximas páginas.

Um grande abraço e força nos estudos!

Prof. Rômulo Santos.

## 1) História do GNU/LINUX:

Em 1976 Ken Thompson solicitou ao *Bell Labs* uma licença de seis meses e foi dar aulas na Universidade de Berkeley, no Estado americano da Califórnia. Lá ele ensinou o Unix e desenvolveu o que viria a ser depois a versão 6, uma versão voltada para universidades. O sistema foi um sucesso e se espalhou rapidamente pelas universidades norte-americanas.

**Unix**<sup>®</sup>  
Operating System

Após a volta de Thompson para o *Bell Labs*, estudantes e professores continuaram a desenvolver o Unix, cujo código era aberto e permitia mudanças, mediante uma licença universitária criada para tal fim. Surgiu o *Berkeley Software Distribution* (BSD), um Unix "totalmente adaptado ao ambiente acadêmico". Conforme disse Ken Thompson: "um sistema operacional feito por programadores para programadores".



The original BSD daemon appeared first in 1983 on the cover of the 4.2BSD manuals published by the Userix Association.



Ken Thompson e Dennis Ritchie considerados os pais do UNIX.

Em 1986 Andrew Stuart Tanenbaum lança o sistema operacional Minix (*mini-unix*). Tratava-se de um sistema pequeno e projetado para o “ensino da disciplina de sistemas operacionais em cursos de graduação”.



O Minix de Tanenbaum

O autor e professor foi constantemente solicitado para adição de novas funcionalidades no sistema. Cabe ressaltar que os sistemas UNIX exigiam o pagamento de uma licença de uso.

Já em 03 de julho de 1991, Linus Torvalds, aluno de graduação da Universidade de Helsinque (Finlândia), implementou “uma versão do Minix como projeto pessoal”. O código-fonte foi distribuído por Linus na Internet (servidor FTP *ftp.funet.fi*) com intuito de encontrar colaboradores que começaram a usar, modificar e criar coisas novas.

Surgiu na época um número elevado de interessados que em conjunto desenvolveram um *kernel* (núcleo do SO). Foi estabelecido um novo modelo de desenvolvimento com as seguintes características:

- Distribuído.
- Colaborativo.
- Com qualidade.
- Dinâmico.
- Eficiente.



Linus Torvalds autor do Linux

**1991 (14 Mai): liberada a versão 0.01 do Linux = Linux + Unix**

- Em Dez 1991, foi lançada a versão 0.11 (primeira a ser auto hospedada).
- Não possuía suporte à rede e suporte a periféricos era extremamente limitado.
- Mecanismo de memória virtual era restrito.
- Sistema de arquivos suportado: *Minix File System*.
- Inicialmente, Linux era monolítico: todo o sistema estava dentro do *kernel*.
- Código possuía 9300 linhas de código C e 950 em *assembly*.
- Era utilizado inicialmente em PC 80386.
- Mesmo tendo código *assembly* misturado aos procedimentos de linguagem C, o sistema foi rapidamente portado para outras plataformas.
- Na versão de 0.12 de fev 1992, Torvalds adotou a licença GPL (*General Public License*) no lugar da anterior, escrita por ele e que não permitia redistribuição comercial.

**1994 (14 Mar): lançada a versão 1.0 do Linux**

- Maior *release* desde o lançamento do sistema.
- O sistema contava com, aproximadamente, 176.250 linhas de código.
- Novo sistema de arquivos.
- Arquivos mapeados em memória.
- Suporte à rede (pilha TCP/IP baseada nos sistemas BSD).
- Muitos aplicativos Unix foram portados para Linux (suporte do GCC).

- Torvalds passou a coordenar as colaborações feitas pelos programadores.
- Somente suportava "sistemas monoprocesso" baseados na i386. A portabilidade se tornou uma preocupação e logo, na versão 1.2, lançada em 07 de março de 1995, ganhou suporte as "arquiteturas Alpha, SPARC e MIPS".

#### **1996 (09 Jun): lançada a versão 2.0 do Linux**

- Sistema passou a ter 470.000 linhas de código C e 8000 linhas de *assembly*.
- Suporte às arquiteturas de 64 bits e Multiprocessamento Simétrico (SMP).
- Novos protocolos de rede: AppleTalk, AX.25, ISDN.
- Montagem remota de Netware e SMB (Samba).
- Manipulação de dependência nos módulos de *kernel*.
- Threads de *kernel*.
- Suporte a diversos periféricos (*device drivers*).
- Foi incluído suporte para mais arquiteturas de processadores.

#### **1999 (26 Jan): lançada a versão 2.2 do Linux**

- Forneceu suporte melhorado para multiprocessadores, adicionou suporte para as arquiteturas "m68k e PowerPC".
- Novos sistemas de arquivos, incluindo suporte somente de leitura para o NTFS da Microsoft.

#### **2001 (04 Jan): lançada a versão 2.4.0 do Linux**

- Suporte para Ligar e Usar (*Plug and Play*) ISA, USB, e PC Cards.
- Suporte ao processador "PA-RISC" da *Hewlett-Packard*.
- Suporte a *Bluetooth*.
- *Logical Volume Manager* - LVM (Gerenciador Lógico de Volumes).
- Suporte RAID e sistemas de arquivo Ext3.

#### **2003 (18 Dez): lançada a versão 2.6.0 do Linux**

- Integração do Microcontrolador Linux  $\mu$ Linux (é pronunciado "iu-si-Linux") no código-fonte principal, suporte a Extensão de Endereço Físico (*Physical Address Extension* - PAE).

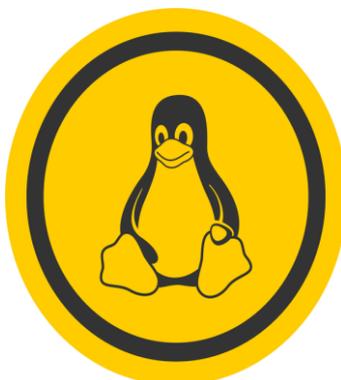
- Integração da Arquitetura Avançada de Som do Linux (*Advanced Linux Sound Architecture* - ALSA).
- Suporte para até 232 usuários.
- Suporte para até 229 IDs de processo.
- Aperfeiçoou o suporte a 64-bits.
- Suporte para sistema de arquivos de até 16 terabytes.
- Preemptividade no escalonamento.

**2003 (18 Dez): lançada a versão 2.6.0 do Linux**

- Suporte para a biblioteca de linhas de execução nativa do POSIX.
- “Integração” do modo de usuário ao código principal do Linux.
- Integração do SELinux ao código principal do Linux.
- Suporte à Infiniband, e muito outros protocolos.
- Em destaque estão as adições de vários sistemas de arquivos através dos lançamentos 2.6.x: FUSE, JFS, XFS, ext4 e outras.

**2011 (22 Jul): lançada a versão 3.0 do Linux**

- Não houve grandes mudanças no núcleo.
- A versão 3.0 foi dada em “comemoração aos 20 anos do Linux”. A partir desta versão foi alterada a forma de interpretação dos números de versionamento do núcleo Linux.



Versão	Mantenedor	Lançamento	Projeção de fim de vida
4.4	Greg Kroah-Hartman	Início de 2016	Fev/2018
4.1	Greg Kroah-Hartman	21/06/2015	Set/2017
3.18	Sasha Levin	07/12/2012	Jan/2017
3.14	Greg Kroah-Hartman	30/3/2014	Ago/2016
3.12	Jiri Slaby	03/11/2013	2016
3.10	Greg Kroah-Hartman	30/6/2013	No final de 2015
3.4	Li Zefan	20/5/2012	Set/2016
3.2	Ben Hutchings	04/01/2012	Mai/2018

**Numeração das Versões:**

No passado, como nas versões 2.4 e 2.6 tínhamos números extensos demais que conhecíamos como X.Y.Z em que:

- X era o número da versão.
- Y tendo número par representava a versão estável e ímpar a versão em desenvolvimento.
- Z representava o *patch level* (ou seja, a versão de lançamento).

Depois ainda, havia um último número em que eram os *fixes* realizados ao longo do tempo. A última versão do 2.6 foi o 2.6.39.4.

Ao final dos últimos lançamentos do *kernel* 2.6, a comunidade resolveu realizar voto para decidir se iriam adotar a versão 2.8 ou 3.0, e por fim, adotaram a versão 3.0. Com isso o conceito X.Y.Z teve alterações em que 3 é a versão estável, o Y a versão de lançamento e Z as correções ao longo do tempo.

**04/01/2012:** versão do *kernel* estável mais antigo ainda suportado: 3.2

**01/04/2016:** último *release* da versão 3.2: 3.2.79

**12/04/2015:** versão estável do *kernel*: 4.0

**08/07/2018:** versão em desenvolvimento do *kernel*: 4.17.5

O *kernel* 4.0 apresentou a possibilidade de se instalar atualizações do *kernel* sem a necessidade de reinicializar o computador. Esse é um recurso bem útil para servidores.

Mais informações, visitem [www.kernel.org](http://www.kernel.org)



**Características Básicas:**

- Multiusuário.
- Multitarefa.
- Time-sharing.
- Genérico.
- Conjunto simplificado de ideias.
- Ortogonalidade entre elementos do sistema.
- Padronização de conceitos fundamentais.



**Objetivos:**

- Simplicidade.
- Generalidade
- Versatilidade.
- Portabilidade.
- Adaptabilidade.



**Inicialmente a comunidade de computação era colaborativa:**

**Hacker:** programador que gosta de escrever *software*.

Denominação é considerada um elogio e, antigamente, era empregada para denominar os melhores programadores de Standford, Carnegie Mellon, MIT e outros grandes centros de pesquisa.

Hoje, o termo é mal-empregado pela imprensa associando o *hacker* a invasores de sistemas e pessoas que cometem crimes através do computador



**Em 1980 Richard Stallman, enfrentou problemas com uma impressora:**

O *hardware* frequentemente engolia papéis.

Papéis ficavam presos e quando os usuários iam buscar suas impressões, constatavam que poucos papéis tinham sido impressos com sucesso.

Ficar ao lado da impressora esperando o seu *job* ser processado era inviável e Stallman modificou o *software* da impressora para que fossem notificadas qualquer situação de *printer jammed* "*Then printer is jammed, please fix it*".



Richard Stallman

**Laboratório do MIT ganhou nova impressora fornecida pela Xerox Corporation:**

Os periféricos se beneficiavam das modificações e *fixes* feitos pelos programadores de centros de pesquisa.

O mesmo problema da impressora anterior foi detectado e Stallman descobriu que existia um pesquisador na Universidade de Carnegie Mellon que estava trabalhando sobre o *software* daquela impressora.

Robert Sproull não repassou o código-fonte do *driver* da impressora por conta de um "Acordo de Não-Divulgação" (*NonDisclosure Agreement* - NDA) assinado com a Xerox.

Em 1984, Stallman deixa o seu trabalho no MIT para atingir o objetivo de fundar e manter uma Comunidade de **Software Livre**.

Criação do projeto GNU: **GNU's Not Unix**.



**As Bases do "Movimento" de Software Livre:** liberdades a serem garantidas:

Executar o *software* para qualquer propósito.

Modificar o programa ajustando-o às suas necessidades ou acrescentando novas características.

**Obs:** "Garantia de acesso ao código-fonte".

Redistribuir o *software*.

Distribuir versões modificadas do programa.

**Obs:** "As versões modificadas devem obedecer aos mesmos princípios".

**Primórdios do "Modelo de Negócios" de Software Livre:**

Projeto deveria iniciar produzindo um sistema operacional (*Hurd*).

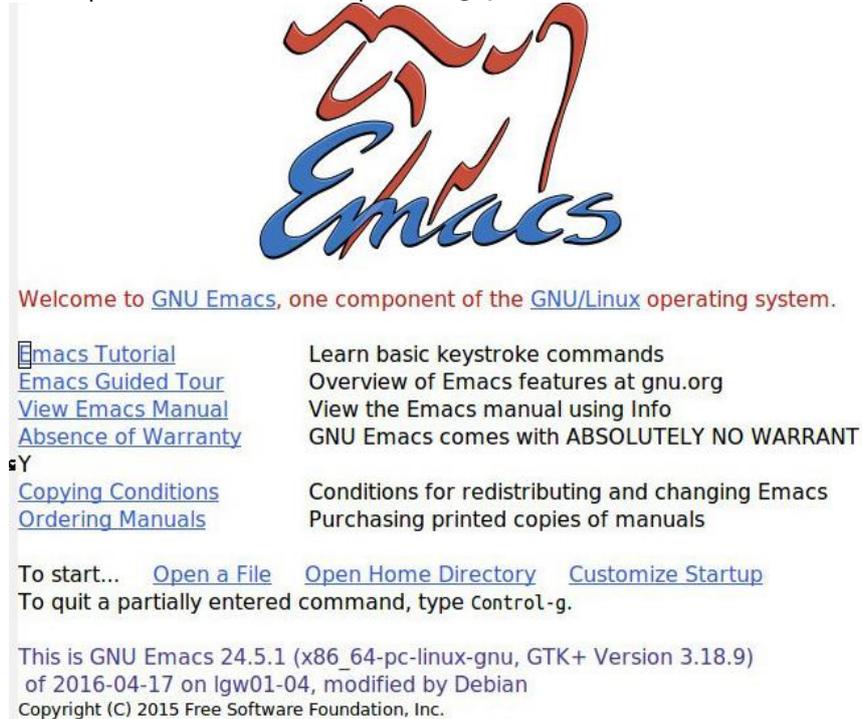
Contudo, não existia compilador livre para tais propósitos, nisso Stallman criou o GCC (GNU C Compiler).

Em 1984 dedica-se à criação do editor Emacs.

Em 1985 editor está estável para utilização.

Muitos programadores solicitavam o *software*, porém alguns usuários não tinham acesso à Internet.

Stallman passou a enviar cópias em fita do Emacs por U\$ 150,00.



Dada a demanda pelo Emacs, em 1985 Stallman cria a FSF (*Free Software Foundation*). Organização sem fins lucrativos cujo objetivo é o de gerenciar o movimento de software livre.

FSF passou a enviar pacotes de *softwares* e manuais. Recebe doações de pessoas e organizações

*Copyleft e GNU GPL*

*Copyleft (L), All Rights Reversed - Don Hopkins.*

*Copyleft utiliza a lei de copyright para legalizar o oposto do seu uso normal: manter o software livre!*

*Copyleft garante os direitos de executar, copiar, modificar e distribuir.*

*Impede de adicionar restrições ao uso do software.*

*Versões modificadas também são livres!*

*GNU GPL (GNU General Public License) é a implementação mais popular do conceito de copyleft.*

### **Software Livre x Open Source:**

Movimento *Open Source* foi fundado por Eric S. Raymond (1998).

Prevê a possibilidade de um programa livre tornar-se um *software* proprietário com direitos autorais (Ex: FreeBSD)

Recebeu grandes incentivos por parte de indústrias.

É comum ocorrer confusão entre os dois movimentos.

Maiores detalhes em <http://www.gnu.org>



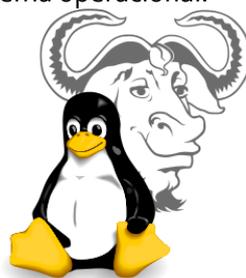
### Por que GNU/Linux?

No início de década de 1990 o *kernel* do GNU/Hurd não estava pronto.

Dada sua característica de *software* livre, o Linux, que havia sido registrado sob a GNU GPL, passou a utilizar as ferramentas desenvolvidas pelo projeto GNU.

A combinação do *kernel* (Linux) com as ferramentas (GNU) gerou o sistema operacional GNU/Linux.

**Obs:** Linux refere-se somente ao *kernel* do sistema operacional.



### Software Livre: Filosofia

GNU GPL *General Public License* (Licença Pública Geral): GNU GPL ou simplesmente GPL, é a designação da licença para *software* livre idealizada por Richard Stallman em 1989, no âmbito do projeto GNU da *Free Software Foundation* (FSF).

A GPL é a licença com maior utilização por parte de projetos de *software* livre, em grande parte devido à sua adoção para o projeto GNU e o SO GNU/Linux.

Segundo a FSF, o Software Livre nos proporciona as seguintes liberdades:

- Liberdade de executar um programa, seja qual for o propósito.
- Liberdade de modificar um programa para adaptá-lo às suas necessidades e, para que isso ocorra, você deve ter acesso ao código-fonte.
- Liberdade de redistribuir cópias, gratuitamente ou mediante uma taxa.
- Liberdade de distribuir versões modificadas do programa e, nesse caso, toda a comunidade poderá beneficiar-se dos aperfeiçoamentos.



A GPL permite que os programas sejam distribuídos e reaproveitados, mantendo, porém, os direitos do autor de forma a não permitir que essa informação seja usada de uma maneira que limite as liberdades originais. A licença não permite, por exemplo, que o código seja apoderado por outra pessoa, ou que sejam impostos sobre ele restrições que impeçam que seja distribuído da mesma maneira que foi adquirido.

“Nossa missão é preservar, proteger e promover a liberdade de usar, estudar, copiar, modificar e redistribuir software, e defender os direitos dos usuários de Software Livre”.

A licença GPL foi publicada em Jan 1989. No entanto, o texto da licença comportava vários problemas. Em Jun 1991 foi publicada a GPL versão 2, sendo ao mesmo tempo introduzida uma nova licença LGPL. Em 2005, Stallman em conjunto com Eben Moglen preparou uma nova versão que foi chamada de GPLv3 e o primeiro esboço foi publicado em Jan 2006, sendo a versão final lançada em Jun 2007.



**Licença BSD:**

Licença de código aberto utilizada nos SO do tipo *Berkeley Software Distribution* (um sistema derivado do Unix). Apesar dela ter sido criada para os sistemas BSD, atualmente vários outros sistemas são distribuídos sob esta licença.

Os proprietários originais da distribuição BSD eram os "Regentes da Universidade da Califórnia", tendo em vista o BSD ter nascido na Universidade de Berkeley. A licença oficial BSD tem sido revisada desde a sua criação, e inspirou inúmeras variantes utilizadas por outros desenvolvedores de *software*.

Há diferenças entre GPL e licenças permissivas de *software* livre (estilo BSD), que não impõem restrições nas versões modificadas. Enquanto a GPL propõe que a liberdade dos softwares se preserve desde os desenvolvedores até o usuário final, outros acreditam que intermediários devam ter a liberdade de modificar e redistribuí-los comercialmente.



**Outros tipos de licença:**

**Freeware:** é o mesmo que gratuito. Sua utilização não necessita de cobrança e todas as funções do programa podem ser acessadas pelo usuário, sem nenhum tipo de restrição. O tempo de uso pelo usuário nesse tipo de licença também não possui restrições. É distribuído de forma gratuita e permite redistribuição, mas não permite alterações (Código Fechado).

**Shareware:** o *software* é distribuído de forma gratuita e possui algumas limitações de relação ao acesso de funcionalidades ou limite de tempo de utilização, geralmente contado em dias, a partir do momento que o software é instalado. Para ter acesso a essas ferramentas bloqueadas ou usar por tempo indeterminado, o usuário é obrigado a comprar o produto (Código Fechado).

**Trial:** versão de testes que permite o uso por um período de tempo (Código Fechado).

**Demo:** versão de demonstração lançado normalmente antes da data oficial para captar futuros usuários (Código Fechado).



**Software Proprietário:** licença de uso restringe cópia, modificação ou redistribuição sem a autorização do proprietário (Normalmente tem Código Fechado).

**Software Comercial:** *software* desenvolvido com a finalidade de gerar lucro. Pode ter (Código Fechado ou Aberto).

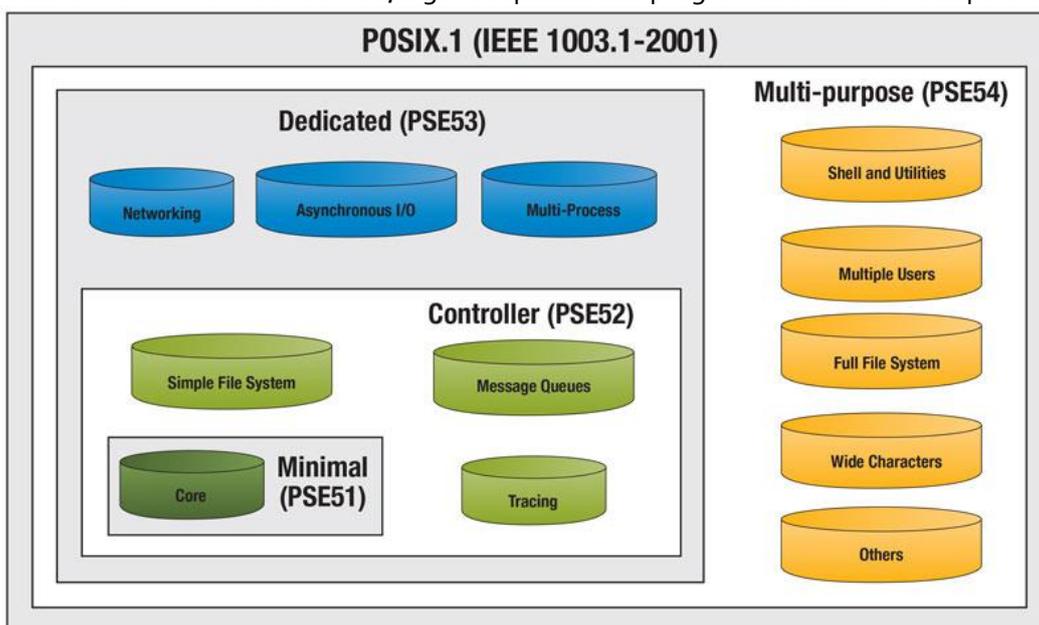
**Domínio Público:** sem proprietário ou direitos reservados (Código fonte pode estar Disponível ou Não).

**Adware:** consiste em inserir uma publicidade no programa, de forma que o usuário é obrigado a ter contato com aquele anúncio todas as vezes que utiliza o *software*. A propaganda só é retirada mediante a um pagamento (Código Fechado).

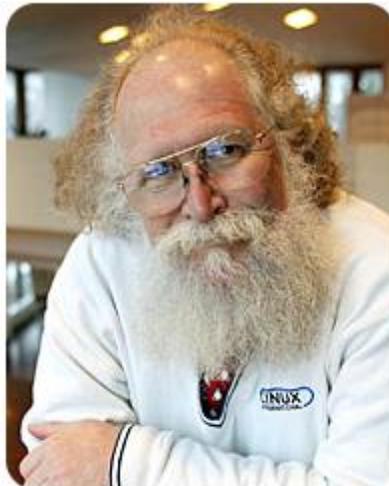
**Open Source:** são totalmente gratuitos e sem nenhum tipo de restrição aos usuários, possuem o código fonte de programação acoplada a eles (Código Aberto).



**Padrão Posix:** *Portable Operating System Interface* a Interface Portável entre Sistemas Operacionais consiste em uma norma concebida pelo *Institute of Electrical and Electronics Engineers* (IEEE), instituição americana criada para definir padrões para equipamentos elétricos e que estabelece a compatibilidade entre SOs. Assim, o Padrão POSIX.1 (norma IEEE 1003.1-2001), é um conjunto de regras que define como um sistema operacional deve funcionar para ser um Unix. Normatiza ainda, alguns aspectos dos programas desenvolvidos para Unix.

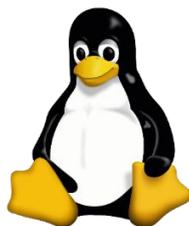


**Linux International (LI):** instituição sem fins lucrativos, sediada nos EUA, que visa à expansão do Linux e da comunidade Linux. A LI reúne vários grupos e faz um trabalho, de nível mundial. O seu presidente, desde 1996, é Jon “Maddog” Hall, que percorre o mundo ministrando palestras para divulgar o Linux e colher resultados da utilização do sistema.



Jon Maddog Hall

**Tux:** o *kernel* Linux possui um logotipo, cujo nome é Tux (fusão de Torvalds com Unix). Em meados de 1996, Larry Ewing apresentou um logotipo da forma como Torvalds queria. Larry desenhou o Tux no GIMP, um famoso *Software Livre* para o tratamento de imagens.



Tux

## 2) Distribuição e características:

### O que é uma distribuição GNU/Linux?

O modelo de *software* livre permite a modificação e adaptação dos aplicativos e utilitários. Já o *kernel* é mantido por uma organização única.

Uma distribuição consiste no encapsulamento do *kernel* adicionado aos aplicativos, mecanismos de instalação e manutenção do sistema. Distribuições distintas podem conter diferenças de configuração e estrutura.

Qualquer pessoa, empresa ou organização pode manter uma distribuição. A figura 1, ilustra diferentes tipos de distribuições GNU/Linux.



Figura 1: Distribuições GNU/Linux.

**Red Hat Linux:** características

Nacionalidade americana. Uma das maiores distribuições do mundo.

Responsável pelo desenvolvimento do gerenciador de pacotes **RPM**.

Facilitou o processo de instalação e manutenção de *softwares* em sistemas GNU/Linux.

Ajustou o mecanismo de compilação e a resolução de dependências.

Adotado por muitas outras distribuições: Conectiva, Mandrake, etc.

Sistema possui agente de atualização para facilitar o processo de *upgrade* do sistema.

Possui um dos melhores suportes para fabricantes de componentes de *hardware*.

Programa de instalação é simples e de fácil utilização.



**3) Shell:**

**Características:**

Programa responsável em interpretar comandos enviados pelo usuário e seus programas ao SO. Faz "interface" entre usuário e *kernel*.

Viabiliza a inicialização e encerramento de programas. Em geral, é o primeiro programa executado após o sucesso no *login*.

Atualmente, sistemas GNU/Linux possuem vários tipos de *shells*.

`$echo $o` (retorna o nome do *shell*, exemplo: *bash*) e `$echo $SHELL` (retorna o caminho (*path*), exemplo: */bin/bash*). A figura 2, ilustra diferentes tipos de *shells* GNU/Linux.

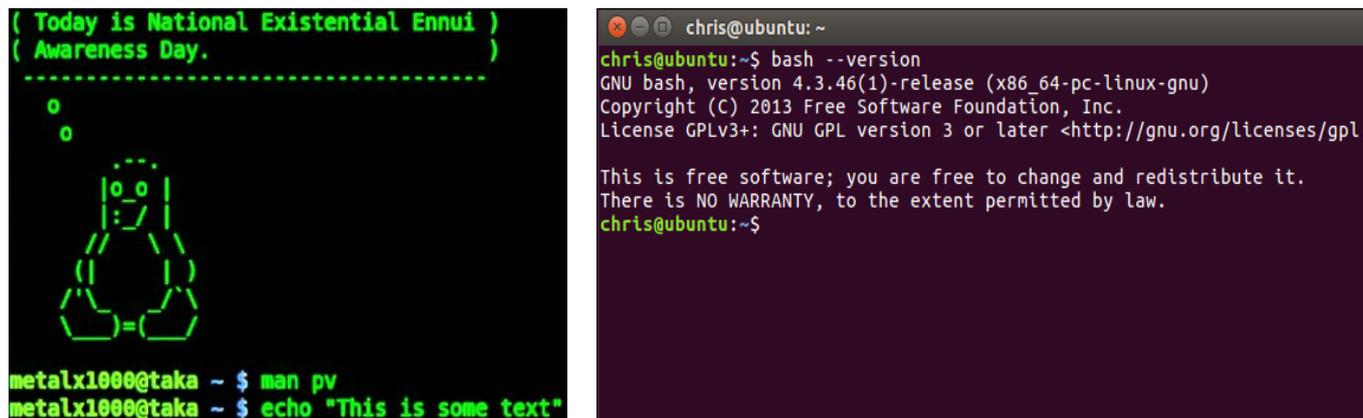


Figura 2: Exemplos de *Shells* GNU/Linux.

**Funções:**

**Bourne Again Shell:** desenvolvido para o GNU/Linux. Possui boa portabilidade e reúne características do *Korn Shell* e *C Shell*. É representado por "*bash*".

Analisar dados a partir do *prompt* (dados de entrada).

Interpretar comandos.

- Ex: **Ctrl+A** ou **Ctrl+X** (início da linha);
- Ctrl+C** (parar a execução de um programa);
- Ctrl+D** (sair do terminal);
- Ctrl+L** (limpar a tela);
- Ctrl+Z** (suspender a execução de um programa).

Controlar ambiente *unix-like* (console).

Fazer redirecionamento de entrada e saída.

Execução de programas.

A figura 3, ilustra a Arquitetura do SO GNU/Linux em Camadas.



Figura 3: Arquitetura do SO GNU/Linux em Camadas.

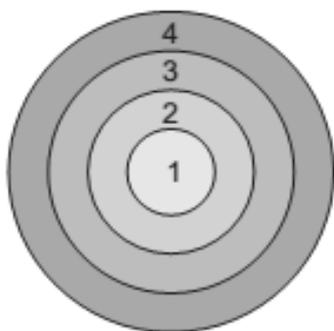
Ao final de cada bloco de aprendizado são apresentadas questões de concursos anteriores para que possamos identificar como o conteúdo costuma ser explorado.

➡ **Observação:** Em caso de dúvidas na resolução das questões recomendo consultar o item "Questões comentadas pelo professor" no final deste material.

## Questões de Fixação

### Questão 01

[ FCC – TRT/11 – 2017 ] A figura abaixo apresenta a típica representação da arquitetura do sistema operacional Linux em camadas, sendo desta forma, conhecida como estrutura shell.



Nessa arquitetura, 1 e 4 são, respectivamente,

- a) Hardware e Programas.
- b) Hardware e Shell.
- c) Kernel e Comandos.
- d) Kernel e Devices.
- e) Kernel e Shell.

### Questão 02

[ FCC – CREMESP – 2016 ] O Bash é interpretador de comandos padrão do sistema operacional Linux e disponibiliza diversos comandos de interação com o usuário. Por exemplo, para limpar a janela do terminal Bash, deve-se pressionar simultaneamente as teclas

- a) Ctrl+d.
- b) Ctrl+a.
- c) Ctrl+l.
- d) Ctrl+c.
- e) Ctrl+x.

### Questão 03

[ FCC – TRT/9 – 2015 ] Um administrador de um computador com sistema operacional Linux está utilizando um terminal Bash para verificar o estado dos processos em execução e pressionou as teclas Ctrl + D. Como resultado do pressionamento dessas teclas

- a) o comando anterior será parado.
- b) o processo anterior será eliminado.
- c) ocorrerá o logout do terminal Bash.
- d) a tela do terminal será limpa.
- e) o cursor será movimentado para o início da linha.

**Questão 04**

[ FCC – TRT/4 – 2015 ] O usuário de um computador com sistema operacional Linux utilizou um terminal shell e executou o xcalc seguido da tecla Enter. Para suspender a execução do xcalc, deve-se, no terminal shell, pressionar simultaneamente as teclas

- a) Alt+z.
- b) Ctrl+z.
- c) Alt+x.
- d) Ctrl+x.
- e) Alt+c.

**GABARITO - Questões de Fixação**

- 1. B
- 4. B

2. A

3. C

**4) FHS:**

**Filesystem Hierarchy Standard:**

Arquivos são organizados de maneira estruturada e hierárquica. O diretório / (barra) é denominado por raiz.

No diretório /home o usuário deve criar sua organização de diretórios pessoais conforme necessidade. É o diretório que o sistema posiciona após o login de um usuário. A figura 4, ilustra a estrutura hierárquica do FHS.

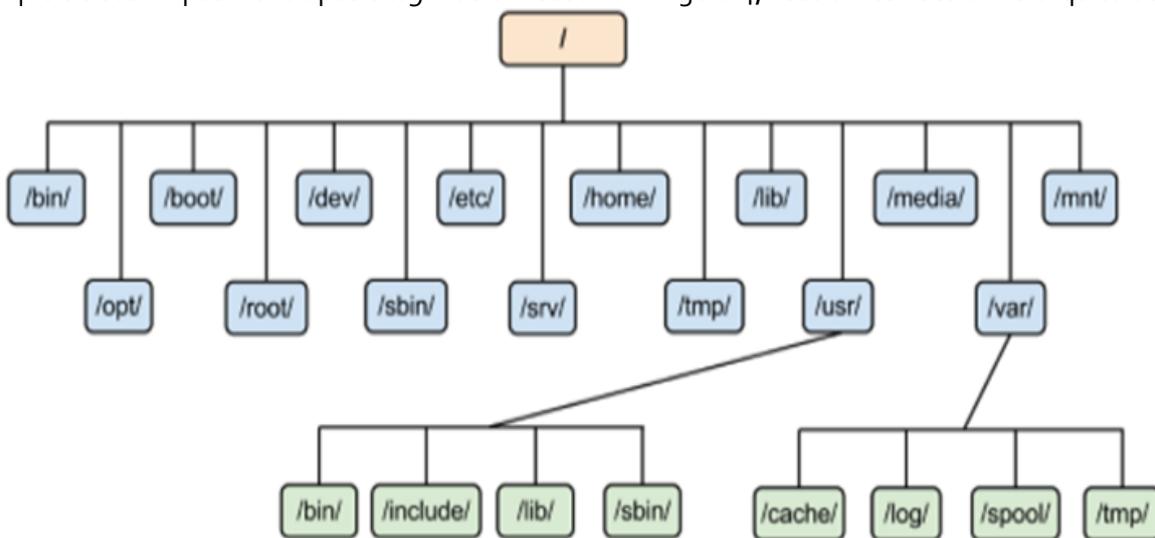


Figura 4: Estrutura Hierárquica do FHS.

É uma padronização da **hierarquia** de **diretórios** dos Unix e derivados. Pretende-se, com isso, criar um **padrão** para que todos os diretórios e arquivos importantes fiquem em locais fixos. Além do FHS, há a **Linux Filesystem Hierarchy** (LFH) que é uma adaptação do FHS ao GNU/Linux.

Está disponível em <http://www.pathname.com/fhs>

- **/bin**: contém executáveis (binários ou não), de uso geral, essenciais em “**recuperação de panes**” do sistema (operação mínima do sistema).
- **/boot**: contém o Kernel Linux e arquivos que controlam a **inicialização** do sistema (gestor de *boot* ou GRUB).
- **/dev**: contém arquivos (*pipes*) de “interligação” do SO a dispositivos de *hardware* (*devices*).
- **/etc**: contém arquivos de “**configuração geral**” do SO e serviços de rede e *scripts* executáveis utilizados na inicialização de serviços.
- **/home**: contém arquivos dos usuários, incluindo as suas configurações particulares.
- **/lib**: contém “módulos do *kernel*, *drivers* e *libraries*” (bibliotecas) essenciais para a **inicialização** do sistema e bibliotecas utilizadas para o funcionamento do executáveis presentes em */bin* e */sbin*.
- **/media**: ponto de montagem para mídias **removíveis** (CD-ROM, DVD, *pendrive*, etc).
- **/mnt**: ponto de montagem para mídias **não removíveis** (HD).
- **/opt**: local destinado a instalação de programas de **terceiros** (ou seja, aqueles não fornecidos pela distribuição).
- **/proc**: diretório (*filesystem*) com conteúdo virtual que controla o sistema (informações colhidas na RAM e geradas constantemente pelo *kernel*).
- **/root**: diretório *home* do **administrador** do sistema.
- **/sbin**: contém executáveis (binários), específicos para a administração do sistema.
- **/sys**: diretório com conteúdo virtual (*sys filesystem*), relacionado ao **controle** do *Kernel* 2.6 ou superior.
- **/tmp**: armazena dados **temporários** de aplicações diversas. O seu conteúdo deverá ser apagado pelo sistema no momento do *boot*. Há o */var/tmp*, cujo conteúdo deverá permanecer após um *reboot*.
- **/usr** (*Unix Shared Resources*): diretório com conteúdo somente leitura e de acesso a todos os usuários. Contém documentos, executáveis não essenciais em momentos de *pane*, *libraries* **não essenciais** etc.
- **/var**: diretório com conteúdo variável, como *logs*, caixa de e-mail e *spool* de impressora.

## Questões de Fixação

### Questão 05

[ FCC – TRE/AP – 2015 ] O técnico judiciário deseja instalar um novo Driver de dispositivo em um computador com sistema operacional RedHat Linux 6. Respeitando-se a estrutura padrão de diretórios, o novo Driver deve ser instalado no diretório

- a) /bin.
- b) /system.
- c) /dev.
- d) /usr/lib.
- e) /etc.

### Questão 06

[ FCC – TRT/15 – 2015 ] O sistema operacional Linux, em suas diferentes distribuições, utiliza uma estrutura de arquivos padronizada e que se baseia na estrutura do Unix. Na estrutura de arquivos do Linux SuSE, o diretório /opt é utilizado

- a) pelo sistema operacional como área de memória secundária quando a memória principal se esgota.
- b) para o armazenamento de logs e eventos gerados pelo sistema operacional.
- c) como a pasta Arquivos de Programas do Windows e é reservado para a instalação de programas adicionais.
- d) para a montagem de dispositivos externos de leitura de mídia óptica.
- e) pelo sistema operacional para o armazenamento de arquivos temporários de gerenciamento de processos.

### Questão 07

[ FCC – TRT/16 – 2014 ] A estrutura de diretórios do sistema operacional Linux possui uma organização padronizada e adotada por todas as distribuições. Considerando que um novo usuário de nome superior seja criado no Linux, o diretório do usuário será criado em:

- a) /root.
- b) /home.
- c) /usr/local.
- d) /tmp.
- e) /usr.

### Questão 08

[ FCC – TRT/12 – 2013 ] Ao contrário de muitos sistemas de arquivo, o ..... não contém nem arquivos de texto nem arquivos binários. Em vez disso ele armazena arquivos virtuais. Desse modo, ele é normalmente citado como um sistema de arquivos virtual. Os arquivos virtuais tipicamente possuem o bytes em seu tamanho, mesmo que contenham um grande número de informações. Em sistemas Red Hat Linux, a lacuna é preenchida corretamente com:

- a) /mnt
- b) /etc

- c) /dev
- d) /proc
- e) /vdir

### GABARITO - Questões de Fixação

5. C  
8. D

6. C

7. B

### 5) Discos e partições:

**Discos IDE (Integrated Drive Electronics):** antigamente, as placas possuíam duas interfaces IDE, uma **primária** e outra **secundária** (HD, unidades de CD-ROM, CD-RW e DVD, etc). No GNU/Linux, elas são identificadas como **hdxy** (*hard disk*), onde "x" é a letra que representa o disco, e "y" é o número da partição.

As figuras 5 e 6, ilustram respectivamente, como o GNU/Linux enxerga os discos e partições IDE, assim como, exemplos de interfaces IDE.

Primeiro disco IDE 0	/dev/hda
Segundo disco IDE 0	/dev/hdb
Primeiro disco IDE 1	/dev/hdc
Segundo disco IDE 1	/dev/hdd

Figura 5: Como o GNU/Linux enxerga os discos e partições IDE.

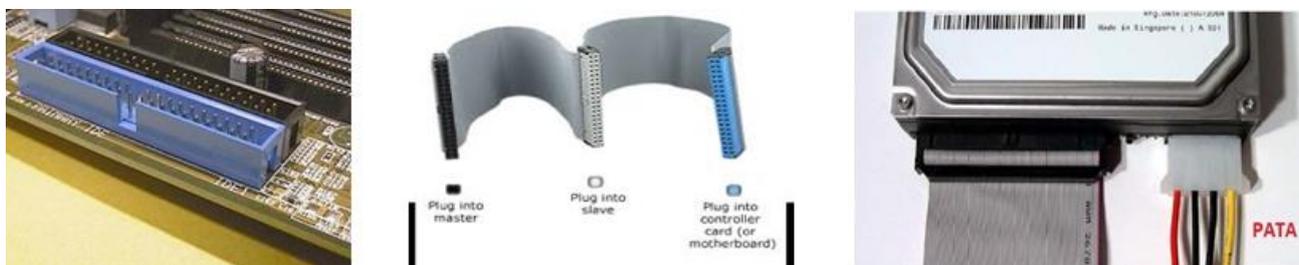


Figura 6: Exemplos de interfaces IDE.

**Discos SCSI (Small Computer System Interface):** são discos mais atuais, utilizados onde a velocidade na leitura e gravação de dados é de maior importância (HD, *pendrives*, unidades de CD-RW, gravadoras de DVD, etc). Essas unidades são referenciadas como **sdxy** (*SCSI disk*), sendo "x" a letra que indica o disco e "y" o número da partição.

**Obs:** Discos SATA, SCSI, SAS e USB são tratados por **sd**.

As figuras 7 e 8, ilustram respectivamente, como o GNU/Linux enxerga os discos e partições SCSI, assim como, exemplos de interfaces SCSI.

Primeiro disco Serial*	/dev/sda
Segundo disco Serial	/dev/sdb
Terceiro disco Serial	/dev/sdc
Quarto disco Serial	/dev/sdd
Unidades CDRom/DVD	/dev/scd0 ou /dev/sr0

Figura 7: Como o GNU/Linux enxerga os discos e partições SCSI.



Figura 8: Exemplos de interfaces SCSI.

**Unidades de disquete:** também conhecidas como "floppy", as unidades de discos flexíveis não são mais utilizadas. Eram referenciadas como fdx, onde "x" é o número da unidade de disquete, começando por "zero".

As figuras 9 e 10, ilustram respectivamente, como o GNU/Linux enxerga os disquetes, assim como, exemplos de disquetes.

- `/dev/fd0` - Primeira unidade de disquetes.
- `/dev/fd1` - Segunda unidade de disquetes.

Figura 9: Como o GNU/Linux enxerga os disquetes.

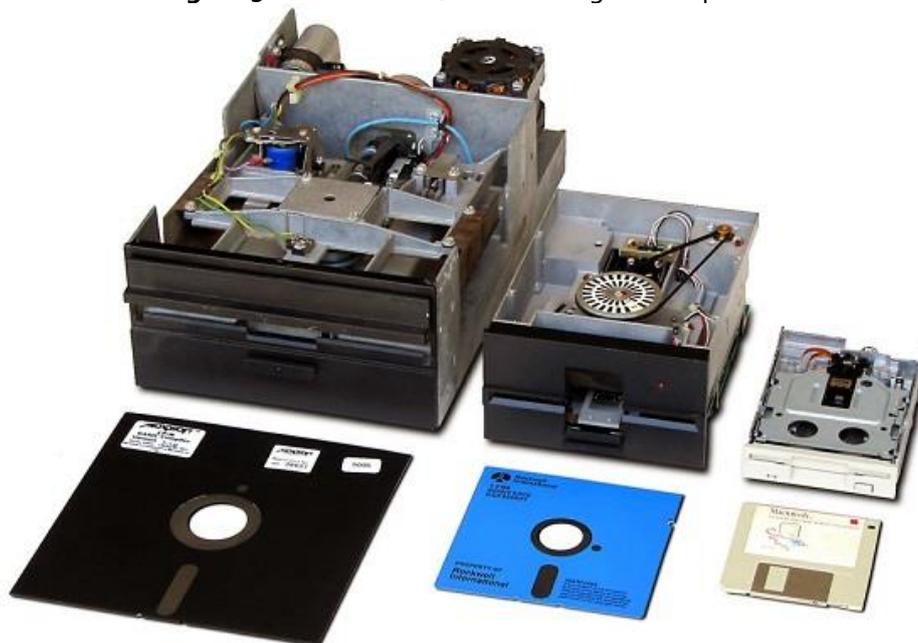


Figura 10: Exemplos de disquetes de 8, 5¼ e 3½ polegadas.

## Questões de Fixação

### Questão 09

[ FCC – CREMESP – 2016 ] Considere a instalação do sistema operacional Linux em um computador do tipo Desktop com processador Intel i7 e um disco rígido padrão SATA de 2TB de capacidade. Nessa situação, o disco que contém a instalação do Linux receberá a denominação de dispositivo:

- a) dev/hda
- b) dev/sda
- c) dev/hdd
- d) dev/sat
- e) dev/hds

## GABARITO - Questões de Fixação

9. B

### 6) MBR e GPT:

**Master Boot Record:** também conhecido como **ms-dos** é o padrão mais **antigo** suportado pela BIOS.

É preciso primeiro **formatar** o disco antes de poder instalar qualquer SO. Existem para isso vários programas de particionamento, como o *qtparted*, *gparted* e outros. Pode ser realizada por intermédio do comando *fdisk*.

Os programas salvam o particionamento na tabela de partição, gravada no início do HD. Esta tabela contém informações sobre o tipo, endereço de início e final de cada partição. Depois do particionamento, vem a formatação de cada partição, onde você pode escolher o sistema de arquivos que será usado em cada uma (EXT3, EXT4, FAT, NTFS, etc).

As figuras 11, ilustra um **Registro Mestre de Inicialização (MBR)**.

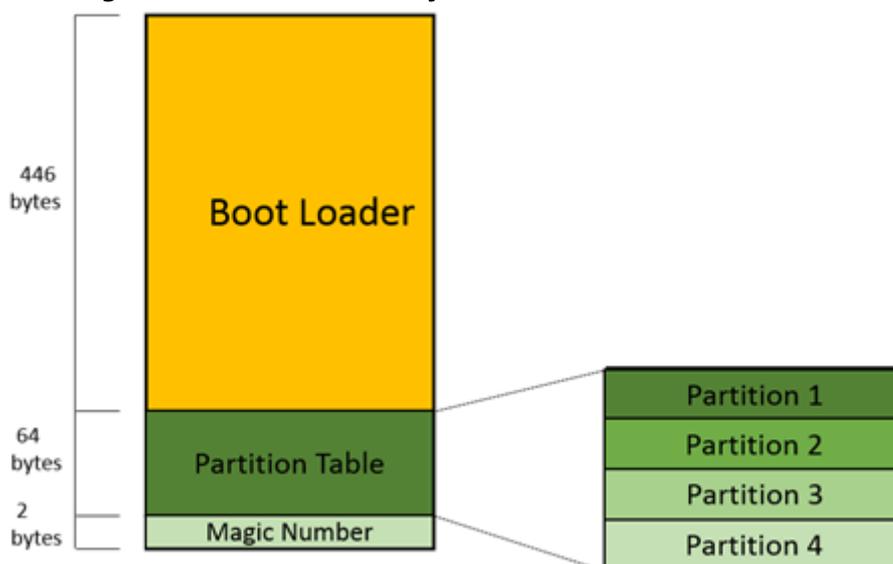


Figura 11: Registro Mestre de Inicialização.

Para manter dois ou mais SO instalados no mesmo computador é necessário instalar um **boot manager**, um pequeno programa que ficará residente na trilha MBR e será carregado cada vez que o computador é ligado. O **boot manager** exibe então uma tela perguntando qual SO deve ser inicializado. A maior parte dos sistemas operacionais atuais trazem algum **boot manager**.

O Windows traz o **NTLDR**, que permite o **dual boot** com versões de Windows, enquanto no Linux temos o **LILO** e o **GRUB**, que podem ser usados para **combinar** Linux e Windows, ou mesmo várias versões diferentes do Linux, de acordo com a configuração. Existem ainda produtos comerciais, como o **Boot Magic**, que faz parte do **Partition Magic**. As figuras 12, ilustra o **Windows Boot Manager (NTLDR)**.

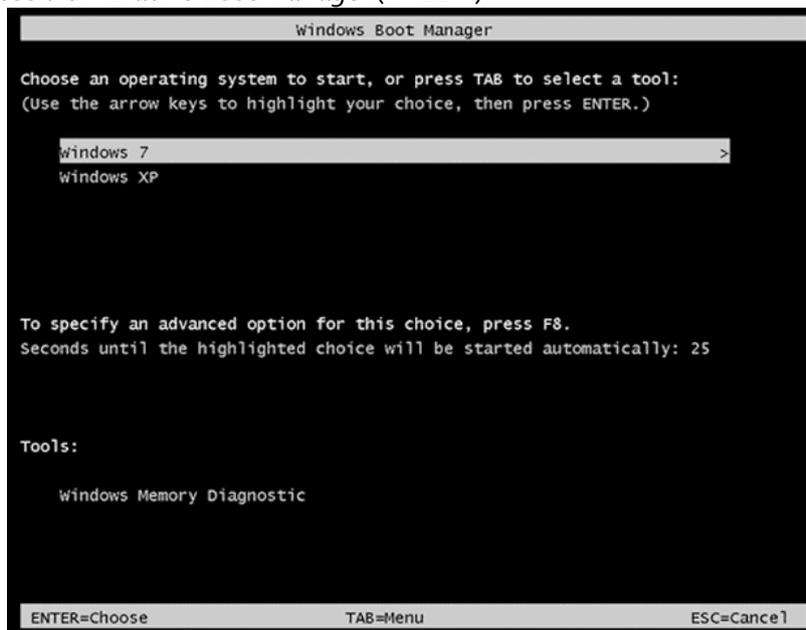


Figura 12: Windows Boot Manager (NTLDR).

As figuras 13, ilustra o gerenciador de **boot GRUB**.

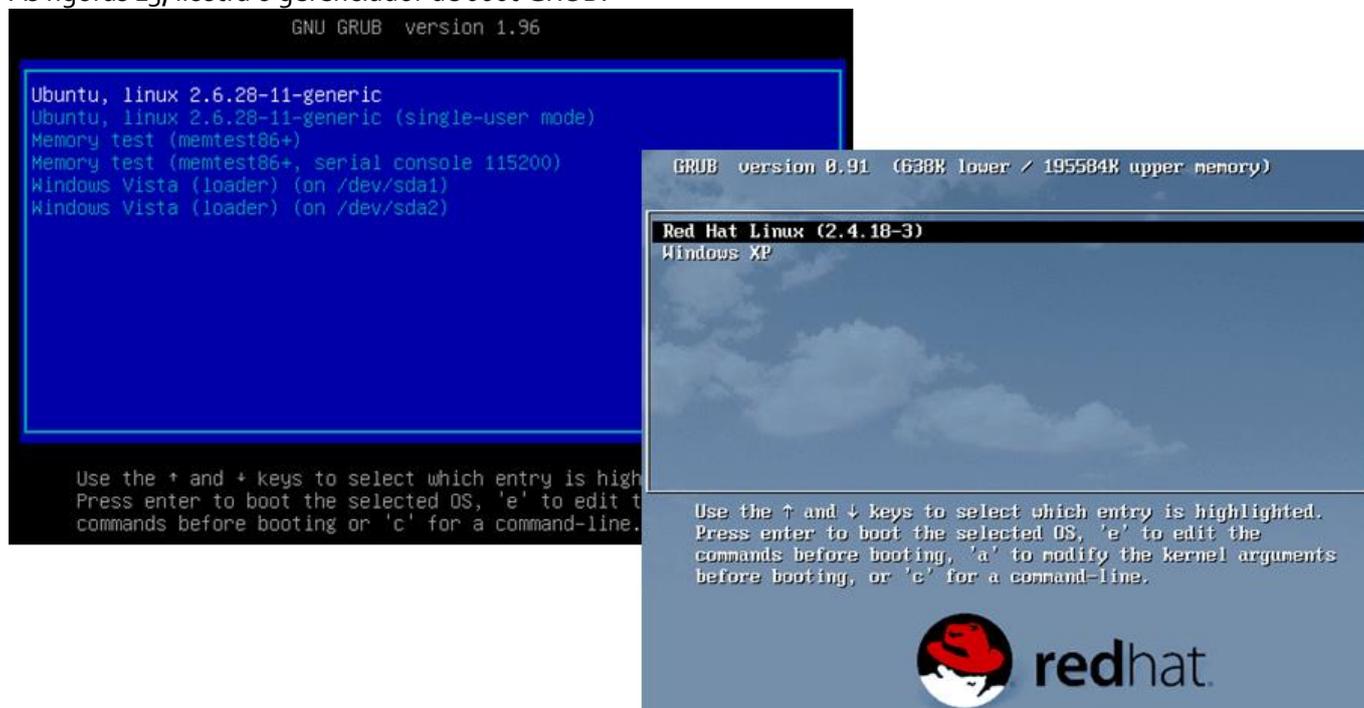


Figura 13: Gerenciador de **boot GRUB**.

**GPT: GUID Partition Table**

O sistema **MBR** foi substituído por **GPT**. O MBR suporta discos com até  $2^{32}$  setores (2 TB). O GPT suporta discos com até 9.4 ZB.

Computadores mais recentes vêm com um *firmware* de substituição para o antigo sistema BIOS chamado **UEFI** (*interface Unified Extensible Firmware*), e o GPT faz parte do padrão UEFI.

GPT suporta até **128** partições primárias. O MBR ocupa 1 setor. O GPT ocupa **34** setores.

O GRUB, que é responsável pela gerência de *boot* no GNU/Linux, normalmente, se instala no MBR ou no GPT.

As figuras 14, ilustra o GPT.

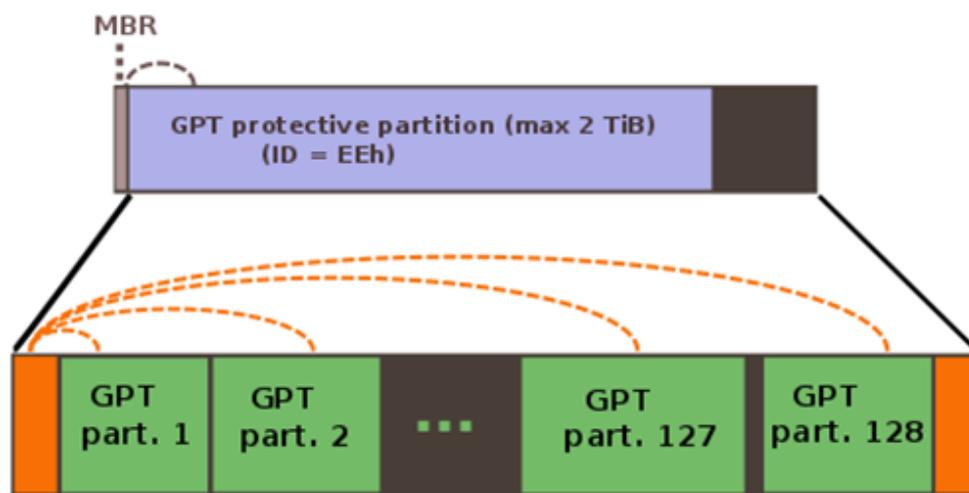


Figura 14: GUID Partition Table.

### Questões de Fixação

**Questão 10**

[ FCC – TRT/9 – 2010 ] Na instalação Red Hat Linux, o gerenciamento de boot (carregamento de inicialização) pode ser configurado pelo LILO ou pelo

- a) LOADLIN.
- b) Boot Manager.
- c) Boot Magic.
- d) Linux Loader.
- e) GRUB.

**Questão 11**

[ FCC – DPE/SP – 2015 ] Muitas distribuições Linux permitem a adição do GRUB no computador durante a instalação do sistema. No entanto, se no ambiente desejado já tem o Linux instalado, o Analista de Suporte, com privilégios de administrador, pode fazer a instalação do GRUB. O GRUB é um

- a) plugin Java para Windows. Contudo, pode ser processado no ambiente Linux, desde que o K3B esteja previamente instalado.

- b) gravador de CD/DVD exclusivo para Linux.
- c) player de vídeo para Linux e Windows.
- d) gerador de boot e trabalha exclusivamente com sistemas Linux.
- e) gerenciador de boot e entre seus principais recursos está a capacidade de trabalhar com diversos sistemas operacionais, como o Linux e o Windows.

## GABARITO - Questões de Fixação

10. E

11. E

### 7) Sistemas de Arquivos:

#### EXT2: *Second Extended File System*

**Vantagens:** é o mais rápido (melhor desempenho) devido ao não uso de journaling, reduz a fragmentação interna permitindo um tamanho de blocos pequeno (gera impacto negativo no desempenho), usa a pré-alocação de blocos (até 8 blocos) quando um arquivo é aberto para gravação, e sempre que possível aloca blocos de forma a reduzir o movimento da cabeça de leitura e gravação.

**Desvantagens:** baixa tolerância a falhas e quantidade pré-determinada de *i-nodes*.

**FSCK** (*File System Check*) ou **SSM** (*System Storage Manager*) são utilitários que permite checar a integridade dos dados em uma partição.

Suporta Arquivos até **2 TB** e Partições até **4 TB**.

#### EXT3: *Third Extended File System*

O sistema de arquivos EXT3 é uma versão do EXT2. Ou seja, tem as mesmas características do EXT2, mas com suporte journaling.

A evolução tornou o EXT3 um sistema de arquivos muito estável e robusto. Suporte a arquivos de até **2 TB** e partições de até **4 TB**.

É possível converter um sistema de arquivos EXT2 para EXT3, adicionado suporte a *journaling*, e também podemos converter um sistema de arquivos EXT3 para EXT2, removendo o suporte a *journaling*.

Suporta os três modos de operação de *journaling*: **Journal**, **Ordered** ou **Writeback**.

Quantidade ilimitada de *i-nodes* (auto incremento).

#### EXT4: *Fourth Extended File System*

Apresenta aprimoramentos quanto a desempenho, escalabilidade e confiabilidade. Mais especificamente, o EXT4 suporta arquivos de até **16 TB** e partições de até **1 EB**.

**Alocação tardia:** a postergação de escrita é um recurso que melhora a alocação de blocos com base no tamanho do arquivo. Isso reduz a fragmentação e melhora o desempenho. Porém, apresenta riscos de ocorrência de perda de dados.

**Jornal *checksumming*:** usa validação (verificação da integridade de dados) do *journaling* melhorando a confiabilidade (mais rápida verificação).

**Uso de extensões:** utilização de um conjunto de blocos contíguos físicos visando melhorar o desempenho e reduzir a fragmentação (fragmentação reduzida).

Compatibilidade com versões anteriores: ext2 e ext3.

Desfragmentação em tempo de execução.

### ReiserFS:

Primeiro sistema de arquivos GNU/Linux com suporte a *journaling*.

Boa performance, principalmente para um número muito grande de arquivos pequenos.

Suporte a arquivos maiores que 2 Gigabytes e o acesso *mhash* (árvore de diretórios) é um pouco mais rápida que o ext3.

Para utilizar *reiserfs* o *kernel* deve possuir "suporte habilitado". O pacote *reiserfsprogs* contém utilitários para formatação.

Organização dos objetos do sistema de arquivos em uma estrutura de dados chamada **B+Trees** (árvores B+).

Alocação dinâmica de *inodes* diminuindo o desperdício de espaço.

Uma desvantagem do *ReiserFS* é seu consumo elevado de **CPU**.

## Questões de Fixação

### Questão 12

[ FCC – TRT/6 – 2018 ] No sistema operacional Red Hat Enterprise Linux 7, uma alternativa para o comando `fsck` é o

- a) `rhck`.
- b) `sfdisk`.
- c) `ssm`.
- d) `fdsk`.
- e) `pfs`.

### Questão 13

[ FCC – TRT/3 – 2015 ] As versões Ext2, Ext3 e Ext4 dos sistemas de arquivos utilizados no Red Hat Linux apresentam a inclusão de novos recursos e a ampliação da capacidade de armazenamento no decorrer da evolução. O que de fato diferencia o Ext2 do Ext3 é a inclusão

- a) da capacidade de formatar e gerenciar adequadamente mídias removíveis como pen drives e cartões SD.

- b) do recurso de alocação do mesmo dado em blocos múltiplos para aumentar a velocidade de acesso ao dado.
- c) da alocação postergada, o que reduz a quantidade de acessos físicos ao disco, reduzindo o tempo de acesso.
- d) do journaling, que aumenta a confiabilidade e elimina a necessidade da checagem do sistema de arquivos após uma parada repentina.
- e) da checagem rápida FSCK sem que haja a necessidade de checar a tabela de alocação.

#### Questão 14

[ FCC – TRT/2 – 2014 ] Considere os seguintes fatores sobre o sistema de arquivos EXT4 no Linux:

- I. O tamanho máximo deste sistema de arquivos é 1EB (exabyte), também suporta arquivos com 16TB (terabytes).
- II. Diretórios podem conter, no máximo, 128.000 subdiretórios, (contrastando com um máximo de 64.000 no ext3).
- III. Não é possível efetuar a montagem de um sistema de arquivos EXT3 existente como sendo um EXT4 sem efetuar seu upgrade. Isto se deve aos diferentes tipos de journaling.
- IV. Neste sistema de arquivos é possível desligar a funcionalidade journaling.

Está correto o que consta APENAS em

- a) II e IV.
- b) II e III.
- c) III e IV
- d) I e II.
- e) I e IV.

#### Questão 15

[ FCC – TRT5 – 2013 ] O EXT3 é um dos sistemas de arquivos mais utilizados no Linux. Este sistema de arquivos

- a) possui apenas dois modos de operação (ordered e writeback), que utilizam endereços de 32 bits e blocos de até 16 KB.
- b) usa o recurso de journaling, que mantém uma lista das alterações realizadas, permitindo que o sistema de arquivos seja reparado de forma muito rápida após, por exemplo, um desligamento incorreto.
- c) reconhece partições de até 2 GB, suporta nomes de arquivos com até 255 caracteres e cada setor possui tamanho fixo de 512 bytes.
- d) usa endereços de 48 bits, o que possibilita endereçar um volume ilimitado de blocos, permitindo criar partições de até 1024 petabytes.
- e) não inclui nenhum sistema de tolerância a falhas, por isso, sempre que o sistema é desligado incorretamente, é necessário utilizar o fsck, para verificar todos os blocos do sistema de arquivos, procurando por erros no disco.

### GABARITO - Questões de Fixação

12. C

13. D

14. E

15. B

### Formatando uma partição de disco:

Você pode formatar uma partição de disco usando o comando **mkfs**.

`mkfs.[filesystem] [opções] [dispositivo]`      `mkfs -t [filesystem] [opções] [dispositivo]`

**filesystem** = ext2, ext3, ext4, msdos, etc.

**opções** =

-F [num] especifica o tipo de FAT 12, 16, e 32.

-n [nome] atribui um nome de volume ao dispositivo.

-c faz a pesquisa por **bad blocks** antes da criação do sistema de arquivo. Os setores defeituosos serão automaticamente marcados para não serem utilizados.

**dispositivo** = identificação da unidade de disco/partição que deseja acessar (como /dev/hda1 (disco rígido) ou /dev/fdo (primeira unidade de disquetes)).

- Formatando um *pendrive* com FAT16:  
`mkfs.msdos -F 16 /dev/sdc`
- Formatando uma partição com EXT3:  
`mkfs.ext3 -n AULAS -c /dev/sda1`

### Montando e acessando uma partição de disco:

Você pode acessar uma partição de disco usando o comando **mount**.

`mount [dispositivo] [ponto de montagem] [opções]`

**dispositivo** = identificação da unidade de disco/partição que deseja acessar (como /dev/hda1 (disco rígido) ou /dev/fdo (primeira unidade de disquetes)).

**ponto de montagem** = diretório de onde a unidade de disco/partição será acessada. O diretório deve estar vazio para montagem de um sistema de arquivo.

**opções** =

-t tipo do filesystem usado pelo dispositivo. Ex: Ext3, Ext4, NTFS, ReiserFS, etc.

-r monta a partição somente para leitura.

-w monta a partição como leitura/gravação.

- Montando uma partição Windows (vfat) de /dev/sda1 em /mnt somente para leitura:  
`mount /dev/sda1 /mnt -r -t vfat`
- Montando um *pendrive* detectado em /dev/sdc em /media:  
`mount /dev/sdc /media -t ext4`

### FSTAB:

Mostra os dispositivos disponíveis no sistema (isto não significa que estejam montados) e é usado pelo **swapon** e **mount** durante o processo de inicialização. O arquivo **/etc/fstab** permite que as partições sejam montadas facilmente especificando somente o dispositivo ou o ponto de montagem. O fstab tem a seguinte forma:

Sistema_de_arquivos	Ponto_de_Mont.	Tipo	Opções	dump	ordem
/dev/sda1	/	ext3	defaults	0	1
/dev/sda2	/boot	ext3	defaults	0	2
/dev/sda3	/dos	msdos	defaults, noauto, rw	0	0
/dev/hdg	/cdrom	iso9660	defaults, noauto	0	0

- **noauto**: não monta os sistemas de arquivos durante a inicialização (útil para CD-ROM).
- **ro**: monta como somente leitura.
- **user**: permite que usuários montem o sistema de arquivos (não recomendado por motivos de segurança).
- **sync**: recomendado para uso com discos removíveis (disquetes, zip drives, etc).
- **defaults**: utiliza valores padrões de montagem que são *rw, exec, auto, nouser, async*.
- **dump**: especifica a frequência de *backup* feita com o programa dump no sistema de arquivos. o (zero) desativa o *backup*.
- **ordem**: define a ordem que os sistemas de arquivos serão verificados na inicialização do sistema. Se usar o (zero), o sistema de arquivos não é verificado.

Após configurar o */etc/fstab*, basta digitar o comando para que a unidade de CD-ROM seja montada.

*mount /cdrom*

**DF:**

Mostra o **espaço livre/ocupado** de cada **partição**.

*df [opções]*

- a: inclui sistemas de arquivos com o blocos.
  - h: mostra o espaço livre/ocupado em MB, KB, GB ao invés de blocos.
  - H: idêntico a -h mas usa 1000 ao invés de 1024 como unidade de cálculo.
  - k: lista em Kbytes.
  - l: somente lista sistema de arquivos locais.
  - m: lista em Mbytes.
  - T: lista o tipo de sistema de arquivos de cada partição
  - t tipo: lista somente sistema de arquivos do tipo definido.
  - x tipo: Não lista sistemas de arquivos do tipo definido.
- Exemplos: *df, df -h, df -t vfat*.

**DU:**

Mostra o **espaço ocupado** por **arquivos** e subdiretórios do diretório atual.

*du [opções]*

- a, --all: mostra o espaço ocupado por todos os arquivos.
  - b, --bytes: mostra o espaço ocupado em bytes.
  - c, --total: faz uma totalização de todo espaço listado.
  - D: não conta links simbólicos.
  - h, --human: mostra o espaço ocupado em formato legível por humanos (Kb, Mb) ao invés de usar blocos.
  - H: como o anterior mas usa 1000 e não 1024 como unidade de cálculo.
  - k: mostra o espaço ocupado em Kbytes.
  - m: mostra o espaço ocupado em Mbytes.
  - S, --separate-dirs: não calcula o espaço ocupado por subdiretórios.
  - x: não faz a contagem de diretórios em sistemas de arquivos diferentes do atual.
- Exemplo: *du -h, du -hc*.

## Questões de Fixação

### Questão 16

[ FCC – MPE/AM – 2013 ] O programa utilizado para construir um sistema de arquivos Linux em um dispositivo (geralmente um disco rígido) é chamado

- a) fdisk.
- b) format.
- c) erase.
- d) fd.
- e) mkfs.

### Questão 17

[ FCC – TRT/23 – 2016 ] Uma partição NFS remota deve ser montada em um computador com sistema operacional Linux. Para especificar, no comando mount, que a partição é NFS, deve-se utilizar a opção:

- a) -n.
- b) -f.
- c) -i.
- d) -s.
- e) -t.

### Questão 18

[ FCC – TRT/15 – 2015 ] O administrador de um servidor com o sistema operacional Linux Red Hat pode utilizar o arquivo `/etc/fstab` para

- a) controlar a montagem dos sistemas de arquivos durante o processo de boot.
- b) definir a tabela de roteamento dos pacotes destinados para a rede de computadores
- c) especificar os sistemas de arquivos utilizáveis pelos usuários padrão.
- d) definir a lista de arquivos disponibilizados por meio do NFS.
- e) listar os sistemas de arquivos disponíveis pela rede de computadores.

### Questão 19

[ FCC – TRF/4 – 2019 ] Usando o ambiente Red Hat Enterprise Linux 7, em condições ideais, para tornar o diretório `/mnt/nfs` disponível como somente leitura para todos os clientes, um Técnico utilizou, corretamente, o comando

- a) `/mnt/nfs *(ro)`.
- b) `systemctl root /mnt/nfs.init`.
- c) `mv /path/to/image.iso mnt/nfs/*.* /basesys/`.
- d) `disp /etc/mnt/nfs`.
- e) `systemctl /mnt/nfs disp`.

**Questão 20**

[ FCC – TRT/15 – 2015 ] O administrador do computador com o sistema operacional Linux SuSE deseja saber o espaço em disco utilizado por um certo usuário. Para obter essa informação, o comando mais adequado para ser utilizado pelo administrador é:

- a) df.
- b) usage.
- c) ls.
- d) utop.
- e) du.

**GABARITO - Questões de Fixação**

16. E

17. E

18. A

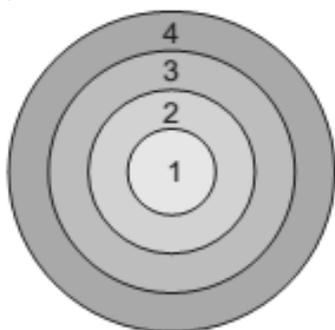
19. A

20. E

**8) Questões comentadas pelo professor:**

**Questão 01**

[ FCC – TRT/11 – 2017 ] A figura abaixo apresenta a típica representação da arquitetura do sistema operacional Linux em camadas, sendo desta forma, conhecida como estrutura shell.



Nessa arquitetura, 1 e 4 são, respectivamente,

- a) Hardware e Programas.
- b) Hardware e Shell.
- c) Kernel e Comandos.
- d) Kernel e Devices.
- e) Kernel e Shell.

**Comentários:**

A arquitetura do sistema operacional Linux é dividida em camadas conforme figura ilustrada a seguir.



Podemos perceber que a parte mais interna é constituída pela base, ou seja, o hardware e a parte mais externa pelo shell e as aplicações. Uma vez que o comando da questão afirmar categoricamente tratar-se do shell as "aplicações" são descartadas como potencial resposta.

**(Resposta B)**

### Questão 02

[ FCC – CREMESP – 2016 ] O Bash é interpretador de comandos padrão do sistema operacional Linux e disponibiliza diversos comandos de interação com o usuário. Por exemplo, para limpar a janela do terminal Bash, deve-se pressionar simultaneamente as teclas

- a) Ctrl+d.
- b) Ctrl+a.
- c) Ctrl+l.
- d) Ctrl+c.
- e) Ctrl+x.

**Comentários:**

Ctrl+L (limpa a tela).

**(Resposta C)**

### Questão 03

[ FCC – TRT/9 – 2015 ] Um administrador de um computador com sistema operacional Linux está utilizando um terminal Bash para verificar o estado dos processos em execução e pressionou as teclas Ctrl + D. Como resultado do pressionamento dessas teclas

- a) o comando anterior será parado.
- b) o processo anterior será eliminado.
- c) ocorrerá o logout do terminal Bash.
- d) a tela do terminal será limpa.
- e) o cursor será movimentado para o início da linha.

**Comentários:**

Ctrl+D (sair do terminal).

**(Resposta C)**

**Questão 04**

[ FCC – TRT/4 – 2015 ] O usuário de um computador com sistema operacional Linux utilizou um terminal shell e executou o xcalc seguido da tecla Enter. Para suspender a execução do xcalc, deve-se, no terminal shell, pressionar simultaneamente as teclas

- a) Alt+z.
- b) Ctrl+z.
- c) Alt+x.
- d) Ctrl+x.
- e) Alt+c.

**Comentários:**

*Ctrl+Z (suspender a execução de um programa).*

**(Resposta B)**

**Questão 05**

[ FCC – TRE/AP – 2015 ] O técnico judiciário deseja instalar um novo Driver de dispositivo em um computador com sistema operacional RedHat Linux 6. Respeitando-se a estrutura padrão de diretórios, o novo Driver deve ser instalado no diretório

- a) /bin.
- b) /system.
- c) /dev.
- d) /usr/lib.
- e) /etc.

**Comentários:**

*/dev: contém arquivos (pipes) de "interligação" do SO a dispositivos de hardware (devices).*

**(Resposta C)**

**Questão 06**

[ FCC – TRT/15 – 2015 ] O sistema operacional Linux, em suas diferentes distribuições, utiliza uma estrutura de arquivos padronizada e que se baseia na estrutura do Unix. Na estrutura de arquivos do Linux SuSE, o diretório /opt é utilizado

- a) pelo sistema operacional como área de memória secundária quando a memória principal se esgota.
- b) para o armazenamento de logs e eventos gerados pelo sistema operacional.
- c) como a pasta Arquivos de Programas do Windows e é reservado para a instalação de programas adicionais.
- d) para a montagem de dispositivos externos de leitura de mídia óptica.
- e) pelo sistema operacional para o armazenamento de arquivos temporários de gerenciamento de processos.

**Comentários:**

*/opt: local destinado a instalação de programas de terceiros (ou seja, aqueles não fornecidos pela distribuição).*

**(Resposta C)**

**Questão 07**

[ FCC – TRT/16 – 2014 ] A estrutura de diretórios do sistema operacional Linux possui uma organização padronizada e adotada por todas as distribuições. Considerando que um novo usuário de nome superior seja criado no Linux, o diretório do usuário será criado em:

- a) /root.
- b) /home.
- c) /usr/local.
- d) /tmp.
- e) /usr.

**Comentários:**

*/home: contém arquivos dos usuários, incluindo as suas configurações particulares.*

**(Resposta B)**

**Questão 08**

[ FCC – TRT/12 – 2013 ] Ao contrário de muitos sistemas de arquivo, o ..... não contém nem arquivos de texto nem arquivos binários. Em vez disso ele armazena arquivos virtuais. Desse modo, ele é normalmente citado como um sistema de arquivos virtual. Os arquivos virtuais tipicamente possuem o bytes em seu tamanho, mesmo que contenham um grande número de informações. Em sistemas Red Hat Linux, a lacuna é preenchida corretamente com:

- a) /mnt
- b) /etc
- c) /dev
- d) /proc
- e) /vdir

**Comentários:**

*/proc: diretório (filesystem) com conteúdo virtual que controla o sistema (informações colhidas na RAM e geradas constantemente pelo kernel).*

**(Resposta D)**

**Questão 09**

[ FCC – CREMESP – 2016 ] Considere a instalação do sistema operacional Linux em um computador do tipo Desktop com processador Intel i7 e um disco rígido padrão SATA de 2TB de capacidade. Nessa situação, o disco que contém a instalação do Linux receberá a denominação de dispositivo:

- a) dev/hda
- b) dev/sda
- c) dev/hdd
- d) dev/sat
- e) dev/hds

**Comentários:**

*A figura abaixo ilustra como o GNU/Linux enxerga os discos e partições.*

Primeiro disco Serial*	/dev/sda
Segundo disco Serial	/dev/sdb
Terceiro disco Serial	/dev/sdc
Quarto disco Serial	/dev/sdd
Unidades CDROM/DVD	/dev/scd0 ou /dev/sr0

(Resposta B)

### Questão 10

[ FCC – TRT/9 – 2010 ] Na instalação Red Hat Linux, o gerenciamento de boot (carregamento de inicialização) pode ser configurado pelo LILO ou pelo

- a) LOADLIN.
- b) Boot Manager.
- c) Boot Magic.
- d) Linux Loader.
- e) GRUB.

**Comentários:**

*O Windows traz o NTLDR, que permite o dual boot com versões de Windows, enquanto no Linux temos o LILO e o GRUB.*

(Resposta E)

### Questão 11

[ FCC – DPE/SP – 2015 ] Muitas distribuições Linux permitem a adição do GRUB no computador durante a instalação do sistema. No entanto, se no ambiente desejado já tem o Linux instalado, o Analista de Suporte, com privilégios de administrador, pode fazer a instalação do GRUB. O GRUB é um

- a) plugin Java para Windows. Contudo, pode ser processado no ambiente Linux, desde que o K3B esteja previamente instalado.
- b) gravador de CD/DVD exclusivo para Linux.
- c) player de vídeo para Linux e Windows.
- d) gerador de boot e trabalha exclusivamente com sistemas Linux.
- e) gerenciador de boot e entre seus principais recursos está a capacidade de trabalhar com diversos sistemas operacionais, como o Linux e o Windows.

**Comentários:**

*O GRUB é um gerenciador de boot que pode ser usado para combinar Linux e Windows, ou mesmo várias versões diferentes do Linux, de acordo com a configuração.*

(Resposta E)

### Questão 12

[ FCC – TRT/6 – 2018 ] No sistema operacional Red Hat Enterprise Linux 7, uma alternativa para o comando fsck é o

- a) rhck.
- b) sfdisk.
- c) ssm.

- d) ffsk.
- e) pfs.

**Comentários:**

*FSCK (File System Check) ou SSM (System Storage Manager) são utilitários que permite checar a integridade dos dados em uma partição.*

**(Resposta C)**

**Questão 13**

[ FCC – TRT/3 – 2015 ] As versões Ext2, Ext3 e Ext4 dos sistemas de arquivos utilizados no Red Hat Linux apresentam a inclusão de novos recursos e a ampliação da capacidade de armazenamento no decorrer da evolução. O que de fato diferencia o Ext2 do Ext3 é a inclusão

- a) da capacidade de formatar e gerenciar adequadamente mídias removíveis como pen drives e cartões SD.
- b) do recurso de alocação do mesmo dado em blocos múltiplos para aumentar a velocidade de acesso ao dado.
- c) da alocação postergada, o que reduz a quantidade de acessos físicos ao disco, reduzindo o tempo de acesso.
- d) do journaling, que aumenta a confiabilidade e elimina a necessidade da checagem do sistema de arquivos após uma parada repentina.
- e) da checagem rápida FSCK sem que haja a necessidade de checar a tabela de alocação.

**Comentários:**

*O sistema de arquivos EXT3 tem as mesmas características do EXT2, mas com suporte journaling.*

**(Resposta D)**

**Questão 14**

[ FCC – TRT/2 – 2014 ] Considere os seguintes fatores sobre o sistema de arquivos EXT4 no Linux:

- I. O tamanho máximo deste sistema de arquivos é 1EB (exabyte), também suporta arquivos com 16TB (terabytes).
  - II. Diretórios podem conter, no máximo, 128.000 subdiretórios, (contrastando com um máximo de 64.000 no ext3).
  - III. Não é possível efetuar a montagem de um sistema de arquivos EXT3 existente como sendo um EXT4 sem efetuar seu upgrade. Isto se deve aos diferentes tipos de journaling.
  - IV. Neste sistema de arquivos é possível desligar a funcionalidade journaling.
- Está correto o que consta APENAS em

- a) II e IV.
- b) II e III.
- c) III e IV
- d) I e II.
- e) I e IV.

**Comentários:**

*Apresenta aprimoramentos quanto a desempenho, escalabilidade e confiabilidade. Mais especificamente, o EXT4 suporta arquivos de até 16 TB e partições de até 1 EB.*

*Journal checksumming: usa validação (verificação da integridade de dados) do journaling melhorando a confiabilidade (mais rápida verificação). Compatibilidade com versões anteriores: ext2 e ext3.*

**(Resposta E)**

**Questão 15**

[ FCC – TRT5 – 2013 ] O EXT3 é um dos sistemas de arquivos mais utilizados no Linux. Este sistema de arquivos

- a) possui apenas dois modos de operação (ordered e writeback), que utilizam endereços de 32 bits e blocos de até 16 KB.
- b) usa o recurso de journaling, que mantém uma lista das alterações realizadas, permitindo que o sistema de arquivos seja reparado de forma muito rápida após, por exemplo, um desligamento incorreto.
- c) reconhece partições de até 2 GB, suporta nomes de arquivos com até 255 caracteres e cada setor possui tamanho fixo de 512 bytes.
- d) usa endereços de 48 bits, o que possibilita endereçar um volume ilimitado de blocos, permitindo criar partições de até 1024 petabytes.
- e) não inclui nenhum sistema de tolerância a falhas, por isso, sempre que o sistema é desligado incorretamente, é necessário utilizar o fsck, para verificar todos os blocos do sistema de arquivos, procurando por erros no disco.

**Comentários:**

É possível converter um sistema de arquivos EXT2 para EXT3, adicionado suporte a journaling, e também podemos converter um sistema de arquivos EXT3 para EXT2, removendo o suporte a journaling.

Suporta os três modos de operação de journaling: Journal, Ordered ou Writeback.

**(Resposta B)**

**Questão 16**

[ FCC – MPE/AM – 2013 ] O programa utilizado para construir um sistema de arquivos Linux em um dispositivo (geralmente um disco rígido) é chamado

- a) fdisk.
- b) format.
- c) erase.
- d) fd.
- e) mkfs.

**Comentários:**

Você pode formatar uma partição de disco usando o comando mkfs.

`mkfs.[filesystem] [opções] [dispositivo]      mkfs -t [filesystem] [opções] [dispositivo]`

**(Resposta E)**

**Questão 17**

[ FCC – TRT/23 – 2016 ] Uma partição NFS remota deve ser montada em um computador com sistema operacional Linux. Para especificar, no comando mount, que a partição é NFS, deve-se utilizar a opção:

- a) -n.
- b) -f.
- c) -i.
- d) -s.
- e) -t.

**Comentários:**

*Você pode acessar uma partição de disco usando o comando mount.*

*mount [dispositivo] [ponto de montagem] [opções]*

*opções =*

*-t tipo do filesystem usado pelo dispositivo. Ex: Ext3, Ext4, NTFS, ReiserFS, etc.*

*-r monta a partição somente para leitura.*

*-w monta a partição como leitura/gravação.*

**(Resposta E)**

**Questão 18**

[ FCC – TRT/15 – 2015 ] O administrador de um servidor com o sistema operacional Linux Red Hat pode utilizar o arquivo /etc/fstab para

- controlar a montagem dos sistemas de arquivos durante o processo de boot.
- definir a tabela de roteamento dos pacotes destinados para a rede de computadores
- especificar os sistemas de arquivos utilizáveis pelos usuários padrão.
- definir a lista de arquivos disponibilizados por meio do NFS.
- listar os sistemas de arquivos disponíveis pela rede de computadores.

**Comentários:**

*Mostra os dispositivos disponíveis no sistema (isto não significa que estejam montados) e é usado pelo swapon e mount durante o processo de inicialização. O arquivo /etc/fstab permite que as partições sejam montadas facilmente especificando somente o dispositivo ou o ponto de montagem.*

**(Resposta A)**

**Questão 19**

[ FCC – TRF/4 – 2019 ] Usando o ambiente Red Hat Enterprise Linux 7, em condições ideais, para tornar o diretório /mnt/nfs disponível como somente leitura para todos os clientes, um Técnico utilizou, corretamente, o comando

- /mnt/nfs \*(ro).
- systemctl root /mnt/nfs.init.
- mv /path/to/image.iso mnt/nfs/\*.\* /basesys/.
- disp /etc/mnt/nfs.
- systemctl /mnt/nfs disp.

**Comentários:**

*mount [dispositivo] [ponto de montagem] [opções]*

*• noauto: não monta os sistemas de arquivos durante a inicialização (útil para CD-ROM).*

*• ro: monta como somente leitura.*

*• user: permite que usuários montem o sistema de arquivos (não recomendado por motivos de segurança).*

*• sync: recomendado para uso com discos removíveis (disquetes, zip drives, etc).*

*• defaults: utiliza valores padrões de montagem que são rw, exec, auto, nouser, async.*

*• dump: especifica a frequência de backup feita com o programa dump no sistema de arquivos. o (zero) desativa o backup.*

*• ordem: define a ordem que os sistemas de arquivos serão verificados na inicialização do sistema. Se usar o, o sistema de arquivos não é verificado.*

**(Resposta A)**

**Questão 20**

[ FCC – TRT/15 – 2015 ] O administrador do computador com o sistema operacional Linux SuSE deseja saber o espaço em disco utilizado por um certo usuário. Para obter essa informação, o comando mais adequado para ser utilizado pelo administrador é:

- a) df.
- b) usage.
- c) ls.
- d) utop.
- e) du.

**Comentários:**

*Mostra o espaço ocupado por arquivos e subdiretórios do diretório atual.*

*du [opções]*

**(Resposta E)**

---

## REFERÊNCIAS:

---

SILVA G, M. **GUIA FOCA GNU/LINUX – Iniciante + Intermediário**. Versão 5.65. Foca GNU/Linux (2010). Disponível em: <<http://www.guiafoca.org/>>.

MOTA, E. **Descobrimo o Linux: Entenda o sistema operacional GNU/Linux**. 3. ed. São Paulo: Novatec Editora, 2012.